Explicit probabilistic models for databases and networks

Tijl De Bie
University of Bristol, Department of Engineering Mathematics,
Bristol, UK
tijl.debie@gmail.com

June 29, 2009

Abstract

Recent work in data mining and related areas has highlighted the importance of the statistical assessment of data mining results. Crucial to this endeavour is the choice of a non-trivial null model for the data, to which the found patterns can be contrasted. The most influential null models proposed so far are defined in terms of invariants of the null distribution. Such null models can be used by computation intensive randomization approaches in estimating the statistical significance of data mining results.

Here, we introduce a methodology to construct non-trivial probabilistic models based on the maximum entropy (MaxEnt) principle. We show how MaxEnt models allow for the natural incorporation of prior information. Furthermore, they satisfy a number of desirable properties of previously introduced randomization approaches. Lastly, they also have the benefit that they can be represented explicitly. We argue that our approach can be used for a variety of data types. However, for concreteness, we have chosen to demonstrate it in particular for databases and networks.

1 Introduction

Data mining practitioners commonly have partial a partial understanding of the structure of the data investigated. The goal of the data mining process is then to discover any additional structure or patterns the data may exhibit. Unfortunately, structure that is trivially implied by the prior information available is often overwhelming, and it is hard to design data mining algorithms that look beyond it. We believe that adequately solving this problem is a major challenge in current data mining research.

For example, it should not be seen as a surprise that items known to be frequent in a binary database are jointly part of many transactions, as this is what should be expected even under a model of independence. Similarly, in random network theory, a densely connected community of high degree nodes should probably be considered less interesting than a similarly tight community among low degree nodes.

Rather than discovering patterns that are implied by prior information, data mining is concerned with the discovery from data of departures from this prior information. To do this, the ability to formalize prior information is as crucial as the ability to contrast patterns with this information thus formalized. In this paper, we focus on the first of these challenges: the task of designing appropriate models incorporating prior information in data mining contexts.

We advocate the formalization of prior information using probabilistic models. This enables one to assess patterns using a variety of principles rooted in statistics, information theory, and learning theory. It allows one to formalize the informativeness of patterns using statistical hypothesis testing—in which case the probabilistic model is referred to as the null model—, the minimum description length principle, and generalization arguments [10, 5, 20, 12, 16, 9, 6].

The most flexible and influential probabilistic models currently used in data mining research have been defined implicitly in terms of randomization invariants. Such invariants have been exploited with success by computationally intensive approaches in estimating the significance (quantified by the p-value) of data mining results with respect to the null model they define [10, 5, 9, 12, 6].

Unfortunately, null models defined in terms of invariants cannot be used to define practical measures of informativeness that can directly guide the search of data mining algorithms towards the more interesting ones. To be

able to do this, *explicit* probabilistic models that take prior information into account, are needed. Applications of models that are defined implicitly in terms of invariants seem to be limited to *post-hoc* analyses of data mining results only.

Despite their potential, the development and use of explicit models is rare in data mining literature, in particular in the study of databases and networks—the main focus of this paper. Furthermore, most of the models that have been proposed elsewhere suffer from serious shortcomings or have a limited applicability (see Discussion in Sec. 5).

In this paper, we present a methodology for efficiently computing explicitly representable probabilistic models for general types of data, able to incorporate non-trivial types of prior information. Our approach is based on the maximum entropy (MaxEnt) principle [8]. Although the methodology is general, for concreteness we focus on rectangular databases (binary, integer, or real-valued) as well as networks (weighted and unweighted, directed and undirected). We further demonstrate remarkably strong connections between these MaxEnt models and the aforementioned randomization approaches for databases and networks.

Outline In the remainder of the Introduction, we will first discuss the maximum entropy principle in general (Sec. 1.1). Then we will discuss how rectangular databases and networks are trivially represented as a matrix (Sec. 1.2), and a way to formalize a common type of prior information for databases and networks as constraints on that matrix (Sec. 1.3). These results allow us to study the design of MaxEnt models for matrices in general, quite independently of the type of data structure it represents (Sec. 2). In Sec. 3, we relate the MaxEnt distributions to distributions defined implicitly using swap randomizations. We then provide experiments demonstrating the scalability of the MaxEnt modelling approach in a number of settings (Sec. 4). In the Discussion (Sec. 5) we point out relations with literature and implications the results in this paper may have for data mining research and applications.

1.1 The maximum entropy principle

Consider the problem of finding a probability distribution P over the data $\mathbf{x} \in \mathcal{X}$ that satisfies a set of linear constraints of the form:

$$\sum_{\mathbf{x}} P(\mathbf{x}) f_i(\mathbf{x}) \quad = \quad d_i. \tag{1}$$

Below, we will show that prior information in data mining can often be formalized as such. For now, let us focus on the implications of such a set of constraints on the shape of the probability distribution. First, note that any probability distribution satisfies the extra conditions

$$\sum_{\mathbf{x}} P(\mathbf{x}) = 1 \quad , \quad P(\mathbf{x}) \geq 0.$$

In general, these constraints will not be sufficient to uniquely determine the distribution of the data. The most common strategy to overcome this problem is to search for the distribution that has the largest entropy subject to these constraints, to which we will refer as the MaxEnt distribution. Mathematically, it is found as the solution of:

$$\begin{aligned} \max_{P(\mathbf{x})} \quad & -\sum_{\mathbf{x}} P(\mathbf{x}) \log P(\mathbf{x}), \\ \text{s.t.} \quad & \sum_{\mathbf{x}} P(\mathbf{x}) f_i(\mathbf{x}) = d_i, \ (\forall i) \qquad (2) \\ & \sum_{\mathbf{x}} P(\mathbf{x}) = 1. \qquad (3) \end{aligned}$$

Originally advocated by Jaynes [7, 8] as a generalization of Laplace's principle of indifference, the MaxEnt distribution can be defended in a variety of ways. The most common argument is that any distribution other than the MaxEnt distribution effectively makes additional assumptions about the data that reduce the entropy. As making additional assumptions biases the distribution in undue ways, the MaxEnt distribution is the safest bet.

A lesser known argument, but not less convincing, is a game-theoretic one [19]. Assuming that the true data distribution satisfies the given constraints, it says that the compression code (e.g. Huffman) designed based on the MaxEnt distribution minimizes the worst-case expected coding length of a message coming from the true distribution. Hence, using the MaxEnt distribution for coding purposes is optimal in a robust minimax sense.

Besides these motivations for the MaxEnt principle, it is also relatively easy to compute a MaxEnt model. Indeed, the MaxEnt optimization problem is convex, and can be solved conveniently using the method of the Lagrange multipliers. Let us use Lagrange multiplier μ for constraint (3) and λ_i for constraints (2). The Lagrangian is then equal to:

$$\begin{aligned} L(\mu, \lambda_i, P(\mathbf{x})) &= -\sum_{\mathbf{x}} P(\mathbf{x}) \log P(\mathbf{x}) \\ &\quad + \sum_i \lambda_i \left(\sum_{\mathbf{x}} P(\mathbf{x}) f_i(\mathbf{x}) - d_i \right) \\ &\quad + \mu \left(\sum_{\mathbf{x}} P(\mathbf{x}) - 1 \right). \end{aligned}$$

Equating the derivative with respect to $P(\mathbf{x})$ to 0 yields the optimality conditions:

$$\log P(\mathbf{x}) = \mu - 1 + \sum_i \lambda_i f_i(\mathbf{x}).$$

Hence, the MaxEnt solution belongs to the exponential family of distributions of the form:

$$P(\mathbf{x}) = \exp\left(\mu - 1 + \sum_i \lambda_i f_i(\mathbf{x}) \right). \tag{4}$$

The Lagrange multipliers should be chosen such that the constraints (3) and (2) are satisfied. These values can be found by minimizing the Lagrangian after substituting Eq. (4) for $P(\mathbf{x})$, resulting in the so-called the dual objective function. After some algebra, we find:

$$\begin{aligned} L(\mu, \lambda_i, P(\mathbf{x})) &= \sum_{\mathbf{x}} \exp\left(\mu - 1 + \sum_i \lambda_i f_i(\mathbf{x}) \right) \\ &\quad - \mu - \sum_i \lambda_i d_i. \end{aligned}$$

This is a smooth and convex function, which can be minimized efficiently using standard techniques (see Sec. 2.2).

1.2 Matrix representations of databases and networks

In this paper, we will apply the MaxEnt principle for the purpose of inferring probabilistic models for rectangular databases as well as networks. Both these data structures can be represented as a matrix, which we will denote as $\mathbf{D}$. This is why they are conveniently discussed within the same paper. Our models will be models for the random matrix $\mathbf{D}$, which is directly useful in modelling databases as well as networks $\mathbf{D}$ may represent.

For the case of databases, the matrix $\mathbf{D}$ has m rows and n columns, and $\mathbf{D}(i, j)$ represents the element at row i and column j. For networks, the matrix $\mathbf{D}$ represents the $n \times n$ adjacency matrix. It is symmetric for undirected networks and potentially asymmetric for directed networks. For undirected networks $\mathbf{D}(i, j) = \mathbf{D}(j, i)$ contains the weight of the edge connecting nodes i and j, whereas for directed networks $\mathbf{D}(i, j)$ contains the weight of the directed edge from node i to node j. In many cases, self-loops would not be allowed, such that $\mathbf{D}(i, i) = 0$ for all i.

To maintain generality, we will assume that all matrix values belong to some specified set $\mathcal{D} \subseteq \Re^+$, i.e.: $\mathbf{D}(i, j) \in \mathcal{D}$. Later we will choose the set $\mathcal{D}$ to be the set $\{0, 1\}$ (to model binary databases and unweighted networks), the set of positive integers, or the set of positive reals (to model integer-valued and real-valued databases and weighted networks). Other choices can be made, and it is fairly straightforward to adapt the derivations accordingly.

For notational simplicity, in the subsequent derivations we will assume that $\mathcal{D}$ is discrete and countable. However, if $\mathcal{D}$ is continuous (such as the set of positive reals) the derivations can be adapted easily by replacing summations over $\mathcal{D}$ with integrals.

1.3 Prior information for databases and networks

For binary databases, it has been argued that row and column sums (also called row and column marginals) can often be assumed as prior information. Any pattern that can be explained by referring to row or column sums in a binary database is then deemed uninteresting. Previous work has introduced ways to assess the significance of data mining results based on this assumption [5, 9, 6]. These methods were based on the fact that the set of databases with fixed row and column sums is closed under so-called swaps.

Swaps are operations that transform any 2×2 submatrix of the form

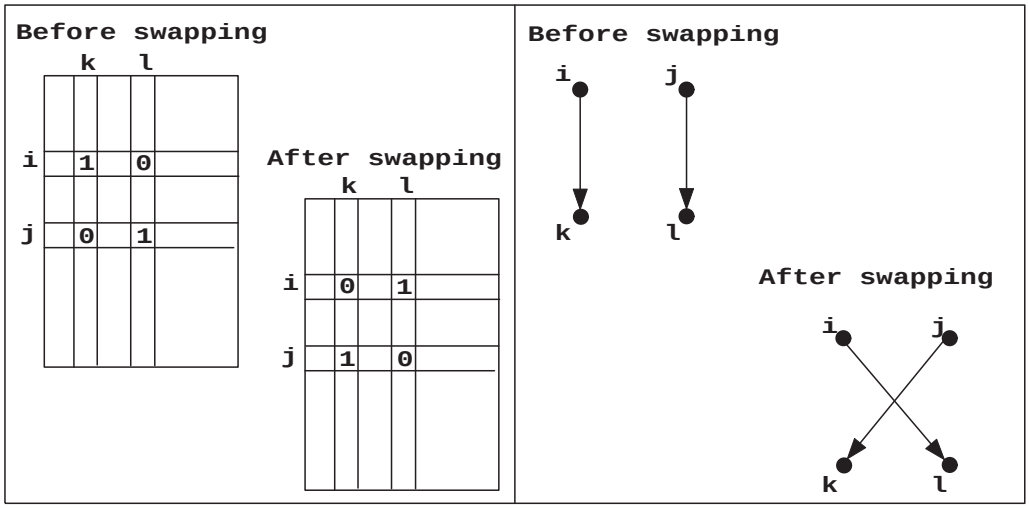

Figure 1: The effect of a swap operation to a binary database (left) and to an unweighted directed network.

$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ into $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. Clearly, such operations leave the row and column sums invariant. Furthermore, it can be shown that by iteratively applying swap randomizations to $\mathbf{D}$, any matrix with the same row and column sums can be obtained. Thus, randomly applying swaps provides a suitable mechanism to sample from the set of databases with fixed column and row sums. See Fig. 1 on the left for a graphical illustration.

The swap operation has later been generalized to deal with real-valued databases as well [12].

Similar ideas have been developed quite independently in the context of network analysis, and in particular in the search for recurring network motifs [10]. There, swaps were applied to the adjacency matrix, corresponding to a rewiring of the kind depicted on the right in Fig. 1. Randomized networks obtained using a chain of such edge swaps were used to statistically assess the significance of particular recurring network motifs in biological networks [10].

Note that the sum of row i of an adjacency matrix $\mathbf{D}$ corresponds to the out-degree of node i, whereas the sum of column j corresponds to the in-degree of node j. Clearly, swaps of this kind leave the in-degree and out-degree of all nodes invariant. I.e., by using swap operations on networks, one

can sample from the set of all networks with given in-degree and out-degree sequences.

In summary, whether $\mathbf{D}$ represents a database or a network, the invariants amount to constraints its row and column sums. The models we will develop in this paper are based on exactly these invariants, be it in a somewhat relaxed form: we will assume that the expected values of the row and column sums are equal to specified values. Mathematically, this can be expressed as:

$$\begin{aligned} \sum_{\mathbf{D}\in\mathcal{D}^{m\times n}} P(\mathbf{D}) \left(\sum_j \mathbf{D}(i,j) \right) &= d_i^r, \\ \sum_{\mathbf{D}\in\mathcal{D}^{m\times n}} P(\mathbf{D}) \left(\sum_i \mathbf{D}(i,j) \right) &= d_j^c, \end{aligned}$$

where d_i^r is the i'th expected row sum and d_j^c the j'th expected column sum. Although they have been developed for binary databases [5] and unweighted networks [10], and later extended to real-valued databases [12], we will explore the consequences of these constraints in broader generality, for various types of databases and weighted networks.

Importantly, it is easy to verify that these constraints are exactly of the type of Eq. (1), such that the MaxEnt formalism is directly applicable.

2 MaxEnt Distributions with Given Expected Row and Column Sums

We have now pointed out that databases as well as networks can be represented using a matrix over a set $\mathcal{D}$ of possible values. Furthermore, commonly used constraints on models for databases as well as for networks amount to constraining the column and row sums of this matrix to be constant. Thus, we can first discuss MaxEnt models for $m \times n$ matrices $\mathbf{D}$ in general, subject to row and column sum constraints. Then we will point out particularities and adjustments to be made for these models to be applicable for matrices representing databases or networks.

The MaxEnt distribution subject to constraints on the expected row and

column sums is found by solving:

$$\begin{aligned}
\max_{P(\mathbf{D})} \quad & -\sum_{\mathbf{D}} P(\mathbf{D}) \log(P(\mathbf{D})), \\
\text{s.t.} \quad & \sum_{\mathbf{D}} P(\mathbf{D}) \left(\sum_{j} \mathbf{D}(i,j) \right) = d_i^r, && (5) \\
& \sum_{\mathbf{D}} P(\mathbf{D}) \left(\sum_{i} \mathbf{D}(i,j) \right) = d_j^c, && (6) \\
& \sum_{\mathbf{D}} P(\mathbf{D}) = 1. && (7)
\end{aligned}$$

The resulting distribution will belong to the exponential family, and will be of the form of Eq. (4):

$$\begin{aligned}
P(\mathbf{D}) &= \exp\left[\mu - 1 + \sum_i \lambda_i^r \left(\sum_j \mathbf{D}(i,j) \right) \right. \\
&\quad \left. + \sum_j \lambda_j^c \left(\sum_i \mathbf{D}(i,j) \right) \right], \\
&= \exp\left[\mu - 1 + \sum_{i,j} \mathbf{D}(i,j)(\lambda_i^r + \lambda_j^c) \right], \\
&= \exp(\mu - 1) \prod_{i,j} \exp\left(\mathbf{D}(i,j)(\lambda_i^r + \lambda_j^c) \right), && (8)
\end{aligned}$$

where μ is the Lagrange multiplier for constraint (7), λ_i^r are the Lagrange multipliers for constraints (5), and λ_j^c for constraints (6).

The first factor in this expression is a normalization constant, the value of which can be determined using constraint (7):

$$\begin{aligned}
\exp(1-\mu) &= \sum_{\mathbf{D} \in \mathcal{D}^{m \times n}} \prod_{i,j} \exp\left(\mathbf{D}(i,j)(\lambda_i^r + \lambda_j^c) \right), \\
&= \prod_{i,j} \sum_{\mathbf{D}(i,j) \in \mathcal{D}} \exp\left(\mathbf{D}(i,j)(\lambda_i^r + \lambda_j^c) \right), \\
&= \prod_{i,j} Z(\lambda_i^r, \lambda_j^c),
\end{aligned}$$

Table 1: Three possible domains for the elements of $\mathbf{D}$, the corresponding normalization factors in the MaxEnt distribution $P(\mathbf{D})$ for the matrix element $\mathbf{D}(i,j)$, and the resulting type of distribution for the matrix elements.

$\mathcal{D}$	$1/Z(\lambda_i^r, \lambda_j^c)$	Distribution
$\{0,1\}$	$1/\left(1+\exp(\lambda_i^r+\lambda_j^c)\right)$	Bernoulli
$\mathbb{N}$	$1-\exp(\lambda_i^r+\lambda_j^c)$	Geometric
$\mathbb{R}^+$	$-(\lambda_i^r+\lambda_j^c)$	Exponential

where $Z(\lambda_i^r, \lambda_j^c) = \sum_{\mathbf{D}(i,j)\in\mathcal{D}} \exp\left(\mathbf{D}(i,j)(\lambda_i^r+\lambda_j^c)\right)$.

Plugging this into Equation (8) yields:

$$\begin{aligned} P(\mathbf{D}) &= \prod_{i,j} \frac{1}{Z(\lambda_i^r, \lambda_j^c)} \exp\left(\mathbf{D}(i,j)(\lambda_i^r+\lambda_j^c)\right), \\ &= \prod_{i,j} P(\mathbf{D}(i,j)), \end{aligned}$$

where

$$P(\mathbf{D}(i,j)) = \frac{1}{Z(\lambda_i^r, \lambda_j^c)} \exp\left(\mathbf{D}(i,j)(\lambda_i^r+\lambda_j^c)\right) \tag{9}$$

is a properly normalized probability distribution for the matrix element $\mathbf{D}(i,j)$ at row i and column j. Hence, the MaxEnt model factorizes as a product of independent distributions for the matrix elements. It is important to stress that we did not impose independence to start. The independence is a consequence of the MaxEnt objective.

Various choices for $\mathcal{D}$ will lead to various distributions, with appropriate values for the normalization constant $Z(\lambda_i^r, \lambda_j^c)$. For $\mathcal{D}$ binary, the MaxEnt distribution for $\mathbf{D}$ is a product of independent Bernoulli distributions with probability of success equal to $\frac{\exp(\lambda_i^r+\lambda_j^c)}{1+\exp(\lambda_i^r+\lambda_j^c)}$ for $\mathbf{D}(i,j)$. For $\mathcal{D}$ the set of positive integers, the distribution is a product of independent geometric distributions with success probability equal to $\exp(\lambda_i^r+\lambda_j^c)$ for $\mathbf{D}(i,j)$. And for $\mathcal{D}$ the set of positive reals, the distribution is a product of independent exponential distributions with rate parameter equal to $-(\lambda_i^r+\lambda_j^c)$ for $\mathbf{D}(i,j)$. This is summarized in Table 1.

2.1 MaxEnt models for databases and networks

The MaxEnt matrix approach is directly applicable for modelling databases of size $m \times n$, and no further modifications are required.

Similarly, for directed networks, given the in-degrees and out-degrees, a MaxEnt model can be found by using the approach outlined above.

Small modifications are needed for the case of undirected networks where $\mathbf{D}(i,j) = \mathbf{D}(j,i)$. From symmetry, it follows that at the optimum $\lambda_i^r = \lambda_i^c$ and we can omit the superscripts r and c, such that the solution looks like:

$$\begin{aligned} P(\mathbf{D}) &= \prod_{i,j} P(\mathbf{D}(i,j)), \\ P(\mathbf{D}(i,j)) &= \frac{1}{Z(\lambda_i, \lambda_j)} \exp\left(\mathbf{D}(i,j)(\lambda_i + \lambda_j)\right), \end{aligned}$$

where $Z(\lambda_i, \lambda_j) = \sum_{\mathbf{D}(i,j) \in \mathcal{D}} \exp\left(\mathbf{D}(i,j)(\lambda_i + \lambda_j)\right)$.

Another small modification is needed for networks where self-loops are not allowed, such that $\mathbf{D}(i,i) = 0$. For $\mathcal{D} \subseteq \Re^+$ these constraints can be enforced quite easily by requiring $\sum_{\mathbf{D}} P(\mathbf{D})\mathbf{D}(i,i) = 0$ for all i, which are again constraints of the form of Eq. (1). The resulting optimal distributions are identical in shape to the model with self-loops, apart from the fact that self-loops receive zero probability and the Lagrange multipliers would have slightly different values, at the optimum.

2.2 Optimizing the Lagrange multipliers

We have now derived the shape of the models $P(\mathbf{D})$, expressed in terms of the Lagrange multipliers (also known as the dual variables), but we have not yet discussed how to compute the values of these Lagrange multipliers at the MaxEnt optimum.

In Sec. 1.1, we have briefly outlined the general strategy to do this, as dictated by the theory of convex optimization [2]: the solution for $P(\mathbf{D})$ in terms of the Lagrange multipliers should be substituted into the Lagrangian of the MaxEnt optimization problem. The result is a smooth and convex function of the Lagrange multipliers, and minimizing it with respect to the Lagrange multipliers yields the optimal values.

Let us investigate what this means for the case of general $m \times n$ matrices. The number of constraints and hence the number of Lagrange multipliers is equal to $m + n + 1$, which is sublinear in the size of the data mn. The

optimal values of the parameters is found easily using standard methods for unconstrained convex optimization such as Newton's method or (conjugate) gradient descent, possibly with a preconditioner [15, 2]. We will report results for two possible choices in the Experiments Section.

In certain cases, the computational and space complexity can be further reduced, in particular when the number of distinct values of d_i^r and of d_j^c are small. Indeed, if $d_i^r = d_k^r$ for specific i and k, the corresponding Lagrange multipliers λ_i^r and λ_k^r will be equal as well, reducing the number of free parameters.

Especially for $\mathcal{D} = \{0, 1\}$, this situation is the rule rather than the exception. Indeed, when the d_i^r and d_j^c are computed based on a given database with m rows and n columns, it is readily observed that both the number of distinct row sums d_i^r and column sums d_j^c are upper bounded by $\min(m, n)$, significantly reducing the complexity when $\min(m, n) \ll \max(m, n)$. Furthermore, the number of different nonzero row sums as well as the number of different nonzero column sums in a sparse matrix with s nonzero elements is at most $\sqrt{2s}$. Hence, the number of dual variables is upper bounded by $2\min(m, n, \sqrt{2s})$. Furthermore, this bound can be sharpened by $d_{\max}^r + d_{\max}^c$ with $d_{\max}^r$ and $d_{\max}^c$ upper bounds on the row and column sums. For $\mathcal{D}$ the set of integers, similar bounds can be obtained.

At first sight, it may seem to be a concern that the MaxEnt model is a product distribution of independent distributions for each $\mathbf{D}(i, j)$. However, it should be pointed out that one does not need to store the value of $\lambda_i^r + \lambda_j^c$ for each pair of i and j. Rather, it suffices to store just the λ_i^r and λ_j^c to compute the probabilities for any $\mathbf{D}(i, j)$ in constant time. Hence, also the space required to store the resulting model is $O(m + n)$, sublinear in the size of the data.

For each of the models discussed in this paper we will make the code freely available.

3 The Invariance of MaxEnt Matrix Distributions to δ-Swaps

We have motivated the use of constraints on the expected row and column sums by relying on previous work where row and column sums of a database or a network adjacency matrix was argued to be reasonable prior information

a data mining practitioner may have about the problem. In this prior work, the authors devised ways to generate new random data satisfying this prior information, by randomizing the given database or network using swaps. These swaps allow one to sample from the uniform distribution over all binary databases with given row and column sums [5], or from all networks with given in-degrees and out-degrees [10]. Later, the swap operation was generalized to allow randomizing a real-valued data matrix as well [12].

We believe the MaxEnt models introduced in this paper are most interesting in their own right, being explicitly represented, easy to compute, and easy to sample random databases or network adjacency matrices from. Still, it is instructive to point out some relations between them and the previously proposed swap operations.

3.1 δ-swaps: a randomization operation on matrices

First, let us generalize the definition of a swap as follows.

Definition 3.1 (δ-swap)**.** Given an $m \times n$ matrix $\mathbf{D}$, a δ-swap for rows i, k and columns j, l is the operation that adds a fixed number δ to $\mathbf{D}(i, k)$ and $\mathbf{D}(j, l)$ and subtracts the same number from $\mathbf{D}(i, l)$ and $\mathbf{D}(j, k)$.

Of course, for a δ-swap to be useful, it must be ensured that $\mathbf{D}(i, j) + \delta, \mathbf{D}(k, j) - \delta, \mathbf{D}(i, l) - \delta, \mathbf{D}(k, l) + \delta \in \mathcal{D}$. We will refer to such δ-swaps as allowed δ-swaps.

Definition 3.2 (Allowed δ-swap)**.** A δ-swap for rows i, k and columns j, l is said to be allowed for a given matrix $\mathbf{D}$ over the domain $\mathcal{D}$ iff $\mathbf{D}(i, j) + \delta, \mathbf{D}(k, j) - \delta, \mathbf{D}(i, l) - \delta, \mathbf{D}(k, l) + \delta \in \mathcal{D}$.

Clearly, an allowed δ-swap leaves the row and column sums invariant. The following Theorem is more interesting.

Theorem 3.1. *The probability of a matrix* $\mathbf{D}$ *under the MaxEnt distribution subject to equality constraints on the expected row and column sums is invariant under allowed* δ*-swaps applied to* $\mathbf{D}$*.*

Indeed, it is easily verified from Eq. (9) that:

$$\begin{aligned} & P(\mathbf{D}(i,j)) \cdot P(\mathbf{D}(i,l)) \\ & \cdot P(\mathbf{D}(k,j)) \cdot P(\mathbf{D}(k,l)) \\ = \; & P(\mathbf{D}(i,j)+\delta) \cdot P(\mathbf{D}(i,l)-\delta) \\ & \cdot P(\mathbf{D}(k,j)-\delta) \cdot P(\mathbf{D}(k,l)+\delta) \end{aligned}$$

for any δ, rows i, k and columns j, l.

This means that for any 2×2 submatrix of $\mathbf{D}$, adding a given number to its diagonal and subtracting the same number from its off-diagonal elements yields the total probability of the data invariant.

More generally, the MaxEnt distribution assigns the same probability to any two matrices that have the same row and column sums. This can be seen from the fact that Eq. (8) is independent from $\mathbf{D}$ as soon as the row and column sums $\sum_j \mathbf{D}(i,j)$ and $\sum_i \mathbf{D}(i,j)$ are given. In statistical terms: the row and column sums are sufficient statistics of the data $\mathbf{D}$. We can formalize this in the following Theorem:

Theorem 3.2. *The MaxEnt distribution for a matrix* $\mathbf{D}$*, conditioned on constraints on row and column sums of the form*

$$\begin{aligned} \sum_j \mathbf{D}(i,j) &= d_i^r, \\ \sum_i \mathbf{D}(i,j) &= d_j^c, \end{aligned}$$

denoted as $P(\mathbf{D}|\sum_j \mathbf{D}(i,j) = d_i^r, \sum_i \mathbf{D}(i,j) = d_j^c)$*, is identical to the uniform distribution over all databases satisfying these constraints.*

This Theorem further strengthens the connection between the uniform distribution over all matrices with fixed row and column sums, as sampled from in [5, 10, 12] using swap randomizations, and the MaxEnt distribution.

3.2 δ-swaps in databases and networks

The invariants that have been used in computation intensive approaches for defining null models for databases and networks are special cases of these more generally applicable δ-swaps.

For binary databases the condition $\mathbf{D}(i,j)+\delta, \mathbf{D}(k,j)-\delta, \mathbf{D}(i,l)-\delta, \mathbf{D}(k,l)+\delta \in \mathcal{D}$ corresponds to the fact that either $\delta = -1$ and $\mathbf{D}(i,k;j,l) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, or $\delta = 1$ and $\mathbf{D}(i,k;j,l) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. Then, the δ-swap is identical to a swap in a binary database. This shows that the MaxEnt distribution of a binary database is invariant under swaps as defined in [5]. For positive real-valued

databases, the δ-swap operations reduce to the Addition Mask method in [12].

Similarly, the edge swap operations on networks can be understood as δ-swaps with δ equal to 1 or -1. For networks in which self-loops are forbidden, swaps with rows i, k and columns j, l must satisfy $\{i, k\} \cap \{j, l\} = \emptyset$, such that no self-loops are created. For undirected networks, a δ-swap operation with rows i, k and columns j, l should always be accomplished by a symmetric δ-swap with rows j, l and columns i, k, in order to preserve symmetry.

Besides these special cases, allowed δ-swaps, found here as simple invariants of the MaxEnt distribution, can be used for randomizing any of the types of databases or networks discussed in this paper. This being said, it should be reiterated that the availability of the MaxEnt distribution should make randomizing the data using δ-swaps unnecessary. Instead one can simply sample directly from the MaxEnt distribution, thus avoiding the computational cost and potential convergence problems faced in randomizing the data. An exception would be if it is crucial that the row and column sums are preserved exactly rather than in expectation.

4 Experiments

In this Section we present experiments that illustrate the computational cost of the MaxEnt modelling approach on a number of problems. All experiments were done on a 2GHz Pentium Centrino with 1GB Memory.

4.1 Modelling binary databases

We report empirical results on four databases: two textual datasets, turned into databases by considering words as items and documents as transactions, and two other databases commonly used for evaluation purposes.

ICDM All ICDM abstracts until 2007, where each abstract is represented by as a transaction and words are items. Stop words have been removed, and stemming performed.

Mushroom A publicly available item-transaction dataset [1].

Pubmed All Pubmed abstracts retrieved by querying with the search query "data mining", after stop word removal and stemming.

Table 2: Some statistics for the databases investigated.

	# items	# tids	support	# closeds
ICDM	4,976	859	5	365,732
Mushroom	120	8,124	812	6,298
Pubmed	12,661	1,683	10	1,249,913
Retail	16,470	88,162	8	191,088

Retail A dataset about transactions in a Belgian supermarket store, where transactions and items have been anonymized [3].

Some statistics are gathered in Table (2). The Table also mentions support thresholds used for some of the experiments reported below, and the numbers of closed itemsets satisfying these support thresholds.

Fitting the MaxEnt model The method we used to fit the model is a preconditioned gradient descent method with Jacobi preconditioner (see e.g. [15]), implemented in C++. It is quite conceivable that more sophisticated methods will lead to significant further speedups, but this one is particularly easy to implement.

To illustrate the speed to compute the MaxEnt distribution, Fig. 2 shows plots of the convergence of the squared norm of the gradient to zero, for the first 20 iterations. The initial value for all dual variables was chosen to be equal to 0. Noting the logarithmic vertical axis, the convergence appears clearly exponential. The lower plot in Fig. 2 shows the convergence of the dual objective to its minimum over the iterations, clearly a very fast convergence in just a few iterations.

In all our experiments we stopped the iterations as soon as this normalized squared norm became smaller than 10^{-12}, which is close to machine accuracy and accurate enough for all practical purposes. The number of iterations required and the overall computation time are summarized in Table 3.

Assessing data mining results Here we illustrate the use of the MaxEnt model for assessing data mining results in the same spirit as [5]. Figure 3 plots the number of closed itemsets retrieved on the original data. Additionally, it shows box plots for the results obtained on randomly sampled databases from

Table 3: The number of iterations required, and the computation time in seconds to fit the probabilistic model to the data.

	# iterations	time (s)
ICDM	13	0.35
Mushroom	37	0.02
Pubmed	15	1.24
Retail	18	2.80

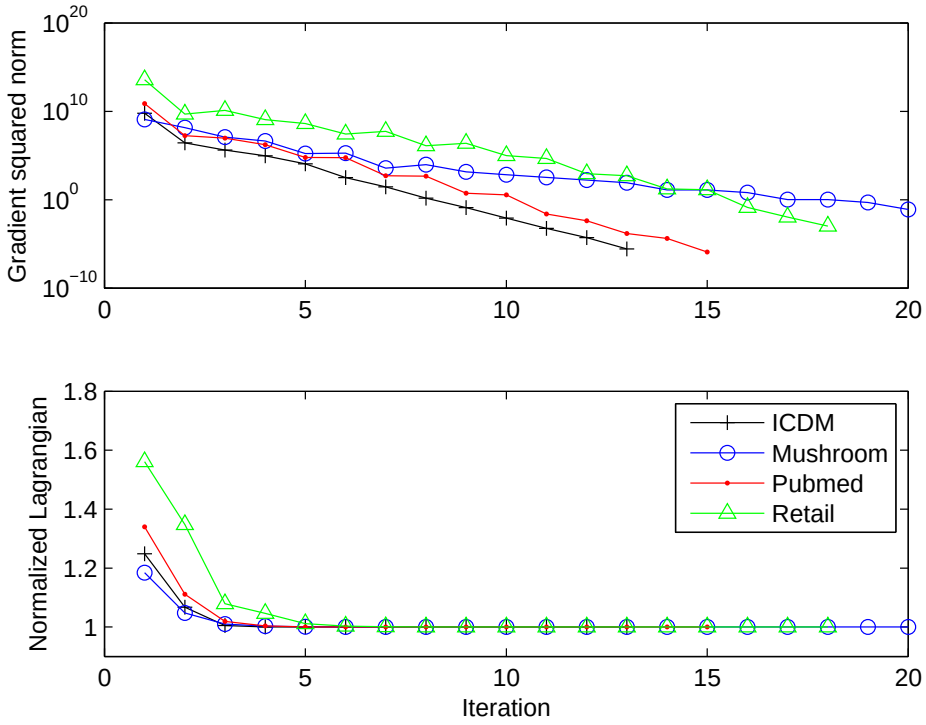

Figure 2: Top: the squared norm of the gradient on a logarithmic scale as a function of the iteration number, plotted for four databases: ICDM abstracts, Mushroom, Pubmed abstracts, and Retail. This plot shows the exponential decrease of the gradient of the dual optimization problem. In the second plot, the convergence of the Lagrange dual is shown for the same databases.

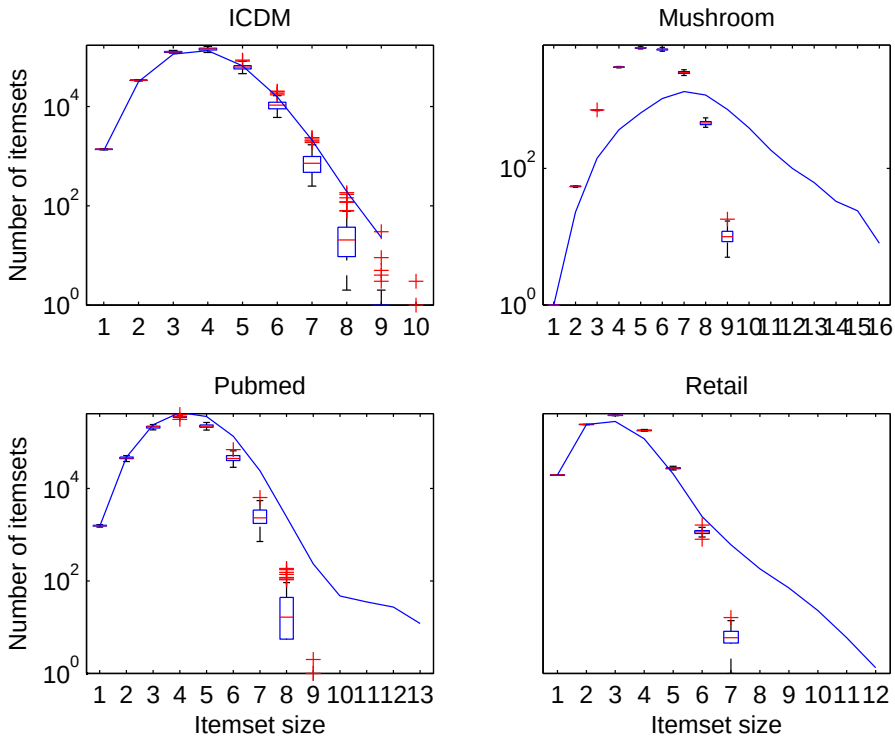

Figure 3: For the four datasets under investigation, these plots show the number of closed itemsets on a logarithmic scale, as a function of their size. Additionally, box plots are shown for the number of closed itemsets as a function of size found on 100 randomized datasets, based the MaxEnt distribution.

the MaxEnt model with expected row sums and column sums constrained to be equal to their values on the original data. If desired, one could extract one global measure from these results, as in [5], and compute an empirical p-value by comparing that measure obtained on the actual data with the result on the randomized versions. However, the plots given here do not force one to make such a choice, and they still give a good idea of the patterns contained in the datasets.

Note that the possible applications of the MaxEnt model will likely reach further than the assessment of data mining results. However, this is beyond the scope of the current paper, and we will get back to this in the Discussion Section.

4.2 Modelling networks

To evaluate the MaxEnt model for networks, we artificially generated power-law (weighted) degree distributions for networks of various sizes between $n = 10$ and $n = 10^6$ nodes, with a power-law exponent of 2.5. I.e., for each n we sampled n expected (weighted) degrees d_i from the distribution $P(d_i) \sim d_i^{-2.5}$. A power-law degree distribution with this exponent is often observed in realistic networks [11], so we believe this is a representative set of examples. However, non-reported experiments on other realistic types of networks yield qualitatively similar results. For each of these degree distributions, we fitted four different types of undirected networks: unweighted networks with and without self-loops, and positive integer-valued weighted networks with and without self-loops.

To fit the MaxEnt models for networks we made use of Newton's method, which we implemented in MATLAB. As can be seen from Fig. 4, the computation time was under 30 seconds even for the largest network with 10^6 nodes. The number of Newton iterations is lower than 50 for all models and degree distributions considered.

This fast performance can be achieved thanks to the fact that the number of different degrees observed in the degree distribution is typically much smaller than the size of the network (see Discussion in Sec. 2.2). The bottom graph in Fig. 4, showing the number of Lagrange multipliers as a function of the network size supports this. The memory requirements remain well under control for the same reasons.

It should be pointed out that in the worst case for dense or for weighted networks (and in particular for real-valued weights), the number of distinct

expected weighted degrees and hence the number of Lagrange multipliers can be as large as the number of nodes n. This would make it much harder to use off-the-shelf optimization tools for n much larger than 1000. However, the problem can be made tractable again if it is acceptable to approximate the expected weighted degrees by grouping subsets of them together into bins, and replacing their values by a bin average. In this way the number of Lagrange multipliers can be brought below 1000. We postpone a full discussion of this and other strategies to a later paper.

5 Discussion

We will first discuss how some existing explicit models are related to particular cases of the MaxEnt models introduced in this paper. Then, we will discuss the implications for data mining of the availability explicit models.

5.1 Connections with literature

Interestingly, the MaxEnt model for binary matrices introduced in this paper is formally identical to the Rasch model, known from psychometrics [13]. This model was introduced to model the performance of individuals (rows) to questions (columns). The matrix elements indicate which questions were answered correctly or incorrectly for each individual. The Lagrange dual variables are interpreted as persons' abilities for the row variables λ_i^r, and questions' difficulties λ_j^c. Remarkably, the model was not derived from the MaxEnt principle but stated directly.

A similar connection exists with the so-called p^* models from social network analysis [14]. Although motivated differently, the p_1 model in particular is formally identical to our MaxEnt model for unweighted networks.

Thus, the present paper provides an additional way to look at these widely used models from psychometrics and social network analysis. Furthermore, as we have shown, the MaxEnt approach suggests generalizations, in particular towards non-binary databases and weighted networks.

Another connection is to prior work on random network models for networks with prescribed degree sequences (see [11] and references therein). The most similar model to the ones discussed in this paper is the one from [4]. In this paper, the authors propose to assume that edge occurrences are independent, with each edge probability proportional to the product of the degrees

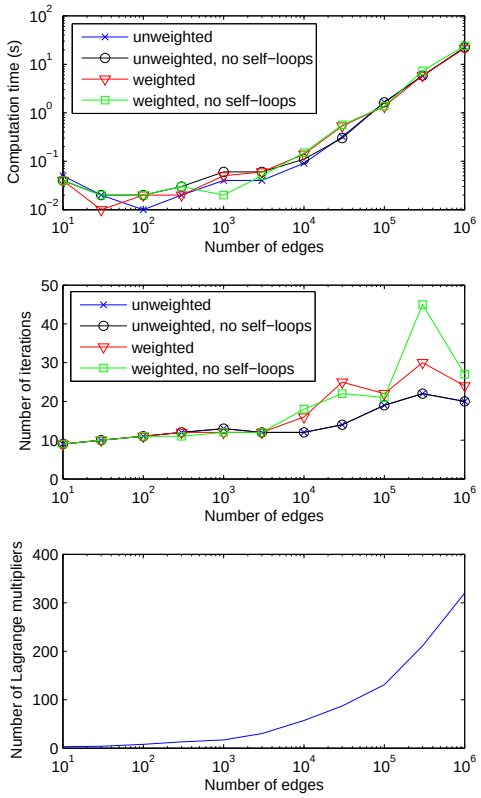

Figure 4: Top: The computation time as a function of the size number of nodes in the network (left). A marker $\times$ is used for the unweighted model with self-loops, $\circ$ for the unweighted model without self-loops, $\triangledown$ for the weighted model with self-loops, and $\square$ for the weighted model without self-loops. Note the log-log scale. Middle: The number of iterations required by the Newton algorithm before convergence. Note the log-scale on the horizontal axis. Bottom: the number of Lagrange multipliers (i.e. the number of variables in the dual of the MaxEnt optimization problem) for the degree sequences investigated, as a function of the network size. Again, note the log-scale on the horizontal axis.

of the pair of nodes considered. In the notation of the present paper:

$$P(\mathbf{D}) = \prod_{i,j} P(\mathbf{D}(i,j)) \quad \text{with} \quad P(\mathbf{D}(i,j)) = \frac{d_i d_j}{s},$$

where $s = \sum_i d_i$. Also for this model the constraints on the expected row and column sums are satisfied.

It would be too easy to simply dismiss this model by stating that among all distributions satisfying the expected row and column sum constraints, it is not the maximal entropy one, such that it is biased in some sense. However, this drawback can be made more tangible: the model represents a probability distribution only if $\max_{i,j} d_i d_j \leq s$, which is by no means true in all practical applications, in particular in power-law graphs. This shortcoming is a symptom of a bias of this model: it disproportionally favours connections between pairs of nodes both of high degree, such that for nodes of too high degrees the edge 'probability' suggested becomes larger than 1. A brief remark considering a similar model for binary databases was made in [5], where it was dismissed by the authors on similar grounds.

5.2 Relevance to data mining

The implications of the ability to construct explicit models for databases and networks are vast. It suffices to look at an area of data mining that has been using data models for a long time in the search for patterns: string analysis.

In the search for patterns in strings, it is common to adopt Markov models of an appropriate order as a background model. Then, patterns that somehow depart from the expected under this background model are reported as potentially interesting. For example, a comprehensive survey about motif discovery algorithms [18] lists various algorithms that model DNA sequences using Markov models, after which motifs are scored based on various information theoretic or statistical measures, depending on the method. One notable example is MotifSampler [17]. Certainly, randomization strategies have been and continue to be used successfully for testing the significance of string patterns. However, it should be emphasized that the methods for discovering string patterns that contrast with a background model have become possible only thanks to the availability of that background model in an explicit form.

Therefore, we believe it is to be expected that the availability of explicit models for databases and networks will enable the introduction of new measures of interestingness for patterns found in such data, be they recurring itemsets or network motifs, other local patterns, or more global patterns such as correlations between items or transactions, the clustering coefficient of a network, and so on.

We also wish to stress that the MaxEnt modelling strategy is applicable for other types of constraints as well. For example, it would be possible to add constraints for the support of specific itemsets, or the presence of certain cliques in a network, etc. Such modifications would be in line with the alternative randomization strategies suggested in [6]. It is also possible to allow matrix elements to become negative, if further constraints for example on the variance are introduced (the resulting MaxEnt distribution would then be a product of normal distributions). Furthermore, the strategy can be applied to other data types as well.

6 Conclusions

In recent years, a significant amount of data mining research has been devoted to the statistical assessment of data mining results. The strategies used for this purpose have often been based on randomization testing and related ideas. The use of randomization strategies was motivated by the lack of explicit models for the data.

We have shown that in a wide variety of cases it is possible to construct explicitly represented distributions for the data. The modelling approach we have suggested is based on the maximum entropy principle. We have illustrated that fitting maximum entropy distributions often boils down to well-posed convex optimization problems. In the experiments, we have demonstrated how the MaxEnt model can be fitted extremely efficiently to large databases and networks in a matter of seconds on a basic laptop computer.

This newfound ability holds several promises for the domain of data mining. Most trivially, the assessment of data mining results may be facilitated in certain cases, as sampling random data instances from an explicit model may be easier than using MCMC sampling techniques from a model defined in terms of invariants of the distribution. More importantly, we believe that our results may spark the creation of new measures of informativeness for patterns discovered from data, in particular for databases and for networks.

References

[1] C. Blake and C. Merz. UCI repository of machine learning databases. In *http://www.ics.uci.edu/mlearn/MLRepository.html*. 1998.

[2] S. Boyd and L. Vandeberghe. *Convex Optimization*. Cambridge University Press, 2004.

[3] T. Brijs, G. Swinnen, K. Vanhoof, and G. Wets. Using association rules for product assortment decisions: a case study. In *Proc. of the 5'th ACM SIGKDD international conference on knowledge discovery in databases*, 1999.

[4] F. Chung and L. Lu. The average distance in a random graph with given expected degrees. *Internet Mathematics*, 1(1):91–113, 2004.

[5] A. Gionis, H. Mannila, T. Mielikinen, and P. Tsaparas. Assessing data mining results via swap randomization. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(3), 2007.

[6] S. Hanhijarvi, M. Ojala, N. Vuokko, K. Puolamaki, N. Tatti, and H. Mannila. Tell me something I don't know: Randomization strategies for iterative data mining. In *Proc. of the 15th ACM SIGKDD conference on knowledge discovery and data mining (KDD09)*, 2009.

[7] E. Jaynes. Information theory and statistical mechanics i. *The Physical Review*, 106:620–630, 1957.

[8] E. Jaynes. On the rationale of maximum-entropy methods. *Proceedings of the IEEE*, 70, 1982.

[9] H. Mannila. Randomization techniques for data mining methods. In *Advances in Databases and Information Systems (ADBIS)*, page 1, 2008.

[10] R. Milo, S. Shen-Orr, S. Itzkovirz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298, 2002.

[11] M. Newman. The structure and function of complex networks. Tech. rep., University of Michigan, 2003.

[12] M. Ojala, N. Vuokko, A. Kallio, N. Haiminen, and H. Mannila. Randomization of real-valued matrices for assessing the significance of data mining results. In *SIAM Data Mining Conference*, pages 494–505, 2008.

[13] G. Rasch. On general laws and the meaning of measurement in psychology. In *Proc. of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, IV*.

[14] G. Robins, P. Pattison, Y. Kalish, and D. Lusher. An introduction to exponential random graph (p^*) models for social networks. *Social Networks*, 29:173–191, 2007.

[15] J. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Tech. rep., CMU, 1994.

[16] A. Siebes, J. Vreeken, and M. van Leeuwen. Item sets that compress. In *SIAM Conference on Data Mining*, 2006.

[17] G. Thijs et al. A higher-order background model improves the detection of promoter regulatory elements by gibbs sampling. *Bioinformatics*, 17(12):1113–1122, 2001.

[18] M. Tompa et al. Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology*, 23(1), 2005.

[19] F. Topsøe. Information-theoretical optimization techniques. *Kybernetika*, 15:8–27, 1979.

[20] G. Webb. Discovering significant patterns. *Machine Learning*, 68(1), 2007.